

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Appellant : Eric A. Benson
Appl. No. : 10/694,509
Filed : October 27, 2003
For : USE OF BROWSER COOKIES
TO STORE STRUCTURED
DATA
Examiner : Jay A. Morrison
Confirmation No. : 2671

APPEAL BRIEF

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Appellant, Appellant in the above-captioned patent application, appeals the final rejection of Claims 1-42 set forth in the final Office Action mailed on August 22, 2006 (hereinafter “the Final Office Action”).

I. REAL PARTY IN INTEREST

The real party of interest in the present application is Amazon.com, Inc.

II. RELATED PROCEEDINGS

No related appeals, interferences or judicial proceedings are currently pending.

III. STATUS OF CLAIMS

Claims 1-42, which are attached hereto as an appendix, are currently pending in the application and are the subject of this appeal. All of these claims stand rejected.

IV. STATUS OF AMENDMENTS

No amendments were made in response to the Final Office Action.

V. SUMMARY OF CLAIMED SUBJECT MATTER

The present application includes four independent claims. Each independent claim is summarized below with citations to corresponding portions of the specification and drawings as required by 37 C.F.R. § 41.37(c)(1)(v). These citations are provided to illustrate specific examples and embodiments of the recited claim language, and are not intended to limit the claims.

Claim 1

Claim 1 is directed to a system (30) for servicing web page requests. The system comprises at least the following:

- a server (36) that responds to user requests for web pages, the server comprising a memory (56) (see, e.g., page 3, ¶ 14; and page 5, ¶ 22);
- schema data (52) stored in the memory of the server, the schema data specifying past and present schemas used to encode data structures into cookies stored on user computers (32) (see, e.g., page 2, lines 24-25; pages 4 and 5, ¶¶ 19-20; and page 6, ¶¶ 24-25);
- a conversion component (50B) executed by the server, the conversion component configured to use the schema data (52) to identify and decode the data structures encoded within cookies received from user computers (32) to generate temporary data structures within the memory of the server (see, e.g., page 2, line 28 to page 3, line 1; page 5, ¶ 22; and page 8, ¶¶ 38-40); and
- application code (40) executed by the server, the application code configured to use the temporary data structures to customize web pages requested by the user computers (see, e.g., page 3, lines 1 and 2; and page 5, lines 23-27).

Claim 18

Claim 18 is directed to a method of extracting information from cookies. The method comprises at least the following:

(a) receiving, at a server (36), cookie data that has a data structure encoded therein, the cookie data received from a user computer (32) (see, e.g., page 2, line 28 to page 3, line 1; page 5, lines 18-21; and page 8, lines 11-13);

(b) determining whether the data structure encoded within the cookie data is valid, wherein the validity of the data structure is determined using information that identifies types of encoded data structures that are currently valid (see, e.g., page 5, lines 20-23; page 8, ¶ 40; and page 41, ¶ 41).

(c) if and only if the data structure encoded within the cookie data is determined to be valid in step (b), decoding the encoded data structure to reproduce the data structure within a memory (56) of the server (see, e.g., page 5, lines 20-25; Fig. 3, block 112; page 8, ¶ 40; and page 9, ¶ 41).

Claim 28

Claim 28 is directed to a method of generating cookie data for storage on a computer of a user. The method comprises at least the following:

- identifying a set of data structures to be encoded within the cookie data (see, e.g., page 5, lines 13-17; Fig. 2, block 78; and page 7, ¶ 33);
- encoding the set of data structures within the cookie data according to schema data stored within a computer memory, the schema data specifying how the set of data structures is to be encoded within the cookie data (see, e.g., page 2, lines 24-27; Fig. 2, blocks 80-90; pages 7 and 8, ¶¶ 33-37; and page 6, lines 6-11); and
- incorporating into the cookie data at least one of the following to facilitate extraction of the set of data structures from the cookie data: (a) a schema identifier, (b) the schema data (see, e.g., page 6, lines 4-6; page 7, lines 25-27; page 9, lines 8-12; and Fig. 2, block 84).

Claim 39

Claim 39 is directed to a method of using browser cookies to store structured data. The method comprises at least the following:

- storing schema data (52) on at least one server computer of a web site system (30), the schema data specifying schemas used by executable software to (a) encode data structures within cookies for storage on user computers (32), and (b) decode the cookies to extract the data structures when the cookies are returned by the user computers (see, e.g., page 2, line 24 to page 3, line 1; and pages 4 and 5, ¶¶ 18-22); and
- modifying the schema data (52) over time to add data structures to, and remove data structures from, a set of data structures encoded within cookies by the executable software (see, e.g., pages 6 and 7, ¶¶ 25-31).

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

The following grounds for rejection are to be reviewed on appeal:

1. The rejection of Claim 1-15, 17-28 and 30-42 under 35 U.S.C. § 102(e) as being anticipated by Durham (U.S. Patent No. 6,330,566);
2. The rejection of Claim 16 under 35 U.S.C. § 103(a) as being unpatentable over Durham in view of McDonough (U.S. Patent No. 5,991,878); and
3. The rejection of Claim 29 under 35 U.S.C. § 103(a) as being unpatentable over Durham in view of Goldberg (U.S. Patent No. 5,907,847).

VII. ARGUMENT

1. Anticipation rejection of Claims Claim 1-15, 17-28 and 30-42 over Durham

As explained below, the anticipation rejection is improper as to each claim because Durham does not expressly or inherently disclose all of the limitations of any claim. *See Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631; 2 USPQ2d 1051, 1053 (Fed. Cir. 1987) (“A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference.”).

Discussion

The present application discloses a general mechanism for encoding data structures in browser cookies. In one embodiment, schema data is used to describe the types of data structures that are eligible for encoding within cookies. This schema data can be generated whenever a new data structure is identified by programmers to include in cookies, and can be added to an existing set of schema data without affecting data describing other encoded data structures. Because the schema data specifies the types of data structures that are to be encoded within cookies at a given point of time, the task of encoding data structures within cookies, and the task of later decoding these cookies, can be performed by executable components that are not specific to any particular type of data structure or set of data structures. Thus, programmers can encode data structures of various types within cookies, without the need to write special code to handle each new type of data structure. For example, a programmer can encode a new type of data structure within a cookie without the need to write special software to break apart and translate that data structure to a cookie-compatible format, and without the need to write special software for regenerating that data structure from received browser cookies.

Durham is directed to a process for encoding user preferences in browser cookies in a compressed format. In contrast to the present application, Durham teaches including specific pieces of preference data one-by-one within cookies. No attempt is made to encode entire data structures within the cookies. This can be seen in the discussion of the encoding process at Figure 5 and col. 10, line 12 to col. 15, line 44, as well as the cookie example in Figure 4. In addition, rather than using schema data, Durham apparently fully specifies the encoding and decoding formats within the executable software that handles the encoding and decoding tasks. (See, e.g., Appendix A of Durham, which demonstrates such a hard-coded implementation.) As a result, a programmer wishing to encode preference data in cookies according to Durham's approach would apparently have to change the encoding and decoding software—or write new software—for each new piece of data to be included.

Claims 1, 3, 11-15 and 17

Appellant is grouping together independent Claim 1, and Claims 3, 11-15 and 17 which depend from Claim 1, for purposes of argument. Claim 1 reads as follows:

1. A system for servicing web page requests, the system comprising:
 - a server that responds to user requests for web pages, said server comprising a memory;
 - schema data stored in the memory of the server, said schema data specifying past and present schemas used to encode data structures into cookies stored on user computers;
 - a conversion component executed by the server, said conversion component configured to use the schema data to identify and decode the data structures encoded within cookies received from user computers to generate temporary data structures within the memory of the server; and
 - application code executed by the server, said application code configured to use the temporary data structures to customize web pages requested by the user computers.

The anticipation rejection of Claim 1 (and thus Claims 3, 11-15 and 17) is improper because, among other reasons, Durham does not disclose "schema data stored in the memory of the server, said schema data specifying past and present schemas used to encode data structures into cookies stored on user computers." As discussed above, Durham does not disclose the encoding of data structures into cookies, or the use of schema data to do so. To the contrary, the process disclosed in Durham involves incorporating individual elements of preference data into a cookie, without regard to the broader structures (if any) from which these data elements are obtained. These data elements represent specific user preferences, and are not "data structures." In addition, as explained above, these data elements are incorporated into the cookie using a hard-coded format (i.e., one that is fully specified within the executable code used to generate the cookie), and not using schema data as claimed. The Final Office Action does not specify what in Durham the Examiner is treating as the "schema data" of Claim 1.

The anticipation rejection of Claim 1 is also improper because Durham does not disclose "a conversion component executed by the server, said conversion component configured to use the schema data to identify and decode the data structures encoded within cookies received from user computers to generate temporary data structures within the memory of the server." In connection with this claim language, the Examiner points to col. 9, line 65 through col. 10, line 11 of Durham. The only portion of this excerpt that involves the decoding of a received cookie is the following sentence: "Code within the response page (an ASP file) disassembles the cookie and

uses its contents to generate the page according to user preferences stored in the cookie.” Nothing in this sentence suggests that Durham’s ASP file uses schema data to identify and decode data structures encoded in cookies, or that the cookie decoding process involves the generation of “temporary data structures.”

In responding to the foregoing arguments regarding the encoding of data structures, the Examiner points out that Claim 5 of Durham refers to a “cookie data structure.” See Final Office Action at page 14, “Response to Arguments” section. The term “cookie data structure,” however, is used in Claim 5 to refer to the cookie as a whole, and not to something that is **encoded in** the cookie. In this regard, even if the cookie as a whole can be considered a “data structure,” it does not necessarily follow that one or more data structures are encoded in the cookie. In connection with this issue, Appellant submits that “data structure” is properly construed in Claim 1 to mean a set of data elements arranged according to an organizational scheme that is not the result of the cookie generation process.

Indeed, nothing in Claim 5 of Durham suggests that a data structure is encoded in the cookie. To the contrary, the only things mentioned in Claim 5 as being encoded in the cookie are individual bits, each of which represents a corresponding client preference. Thus, Claim 5 of Durham does not disclose the decoding of “data structures encoded within cookies” as claimed.

In response to Appellant’s argument regarding the use of schema data, the Examiner points out that Durham discloses incorporating of a “version number” into the generated cookie so that the system can later know what cookie format to expect. See Final Office Action at page 14, “Response to Arguments” section, and Durham at col. 10, lines 12-16. This version number, however, does not imply the use of schema data as claimed. Rather this version number appears to merely identify the hard-coded format used to generate the cookie. In this regard, although Durham may generate cookies according to a schema, the schema is embodied (hard coded) in Durham’s executable code rather than in schema data as claimed. Thus, to modify the types of preference information encoded in the cookies in Durham’s system, a programmer would apparently have to modify the executable code used to generate and decode the cookies.

Because Durham does not disclose all of the limitations of Claim 1 (explicitly or inherently), the anticipation rejections of Claims 1, 3, 11-15 and 17 are improper and should be reversed.

Dependent Claim 2

The anticipation rejection of Claim 2 is improper because Claim 2 depends from Claim 1. In addition, the rejection of Claim 2 is improper because Durham does not disclose the limitations added by Claim 2, namely “wherein the conversion component is not specific to a particular type of data structure.” In connection with these limitations, the Examiner points to col. 4, lines 12-48, which describes generally how various types of data elements may be encoded into a cookie. Even if, *arguendo*, these data elements could somehow be considered “data structures,” the limitations of Claim 2 would not be met because Durham’s conversion component is specific to particular types of data elements. This is because the types of data elements encoded into cookies in Durham are hard coded within the executable code.

Dependent Claim 4

The anticipation rejection of Claim 4 is improper because Claim 4 depends from Claim 1. In addition, the rejection of Claim 4 is improper because Durham does not disclose the limitations added by Claim 4, namely “wherein the schema data specifies the content and format of each of a plurality of data structures that are encoded within the cookies.” The Examiner asserts that Durham discloses these limitations at col. 9, lines 32-65. Appellant respectfully disagrees. The cited portion of Durham summarizes how client preferences are obtained via a user interface, and mentions that these client preferences are both stored in a database and encoded in a cookie. Nothing in this or any other portion of Durham suggests the existence of schema data that “specifies the content and format of each of a plurality of data structures that are encoded within the cookies” as claimed.

Dependent Claim 5

The anticipation rejection of Claim 5 is improper because Claim 5 depends from Claim 1. In addition, the rejection of Claim 5 is improper because Durham does not disclose the limitations added by Claim 5, namely “wherein the schema data includes, for a given data structure that is encoded within cookies, an identification of primitive fields of the given data structure and

datatypes of said primitive fields.” The Examiner asserts that Durham discloses these limitations at col. 4, lines 20-48. Appellant respectfully disagrees. The cited portion of Durham describes how individual values representing client preferences can be concatenated into a bit sequence which is then compressed for inclusion within a cookie. The cited text also mentions the use of a version number to keep track of the cookie format used. Nothing in this portion or any other portion of Durham suggests the existence of schema data that includes an identification of primitive fields of a data structure, or of datatypes of these primitive fields.

Dependent Claim 6

The anticipation rejection of Claim 6 is improper because Claim 6 depends from Claim 5, discussed above. In addition, the rejection of Claim 6 is improper because Durham does not disclose the limitations added by Claim 6, namely “wherein the schema data further includes address offsets of the primitive fields.” The portion of Durham cited by the Examiner, namely col. 4, lines 20-48, does not disclose address offsets of primitive fields, let alone schema data that includes such address offsets.

Dependent Claim 7

The anticipation rejection of Claim 7 is improper because Claim 7 depends from Claim 1. In addition, the rejection of Claim 7 is improper because Durham does not disclose the limitations added by Claim 7, namely “wherein the schema data includes, for a given data structure that is encoded within cookies, an indication of a range of schema versions for which the data structure is valid, wherein the conversion component uses said range to determine whether a particular data structure encoded within a cookie is valid.” The portion of Durham cited by the Examiner (col. 4, lines 12-48) does not disclose schema data that includes an indication of a range of schema versions for which a data structure is valid. In addition, Durham does not disclose the use of a range to determine whether a particular data structure encoded within a cookie is valid. Accordingly, Durham does not disclose the limitations at issue.

Dependent Claim 8

The anticipation rejection of Claim 8 is improper because Claim 8 depends from Claim 1. In addition, the rejection of Claim 8 is improper because Durham does not disclose the limitations added by Claim 8, namely “wherein the schema data includes information about at least one data

structure that is no longer in use.” In connection with this claim, the Examiner points to col. 10, lines 12-16, which mentions that the version number included in a cookie may be used to determine whether the client has become out of date. Nothing in Durham, however, suggests that this determination is made using “schema data [that] includes information about at least one data structure that is no longer in use.” Rather, the determination would presumably be made by simply examining the received cookie’s version number. Thus, the cited portion of Durham does not support the rejection.

Dependent Claim 9

The anticipation rejection of Claim 9 is improper because Claim 9 depends from Claim 1. In addition, the rejection of Claim 9 is improper because Durham does not disclose the limitations added by Claim 9, namely “wherein the conversion component uses the schema data to determine which of a set of the data structures encoded within a received cookie are to be decoded for use.” The portions of Durham cited by the Examiner, namely col. 11, lines 15-36 and col. 15, lines 24-50, illustrate how specific client preferences may be encoded within a cookie. Neither these nor any other portion of Durham discloses either “a set of data structures encoded within a received cookie” or the use of schema data to determine which data structures to decode.

Dependent Claim 10

The anticipation rejection of Claim 10 is improper because Claim 10 depends from Claim 1. In addition, the rejection of Claim 10 is improper because Durham does not disclose the limitations added by Claim 10, namely “wherein the conversion component uses the schema data to determine whether a particular data structure that is encoded within a received cookie is to be decoded for use.” The portions of Durham cited by the Examiner, namely col. 11, lines 15-36 and col. 15, lines 24-50, simply do not disclose this feature.

Claims 18-21, 23 and 25-27

Appellant is grouping together independent Claim 18, and Claims 19-21, 23 and 25-27 which depend from Claim 18, for purposes of argument. Claim 18 reads as follows:

18. A method of extracting information from cookies, the method comprising:
 - (a) receiving, at a server, cookie data that has a data structure encoded therein, said cookie data received from a user computer;

(b) determining whether the data structure encoded within the cookie data is valid, wherein the validity of the data structure is determined using information that identifies types of encoded data structures that are currently valid; and

(c) if and only if the data structure encoded within the cookie data is determined to be valid in step (b), decoding the encoded data structure to reproduce the data structure within a memory of the server.

The anticipation rejection of Claim 18 (and thus Claims 19-21, 23 and 25-27) is improper because, among other reasons, Durham does not disclose “receiving, at a server, cookie data that has a data structure encoded therein.” As discussed above, although the cookies received by Durham’s system include data elements representing user preferences, nothing in Durham suggests that the cookies include encoded data structures. The portions of Durham cited by the Examiner, namely Figure 3; col. 8, line 2 to col. 9, line 32; and col. 10, lines 9-11, do not suggest otherwise. In connection with this issue, Appellant submits that “data structure” is properly construed in Claim 18 to mean a set of data elements arranged according to an organizational scheme that is not the result of the cookie generation process.

The anticipation rejection of Claim 18 is also improper because Durham does not disclose “determining whether the data structure encoded within the cookie data is valid, wherein the validity of the data structure is determined using information that identifies types of encoded data structures that are currently valid.” In connection with this claim language, the Examiner points to col. 8, lines 13-20, where Durham briefly describes what happens if a client transmits a cookie that is incomplete or damaged. While this portion of Durham may imply a capability to determine whether the cookie itself is valid, it does not explicitly or inherently disclose the limitations at issue. In this regard, there is nothing in the reference to suggest that the cookie validation process would involve either “determining whether [a] data structure encoded within the cookie data is valid” or “using information that identifies types of encoded data structures that are currently valid.”

The anticipation rejection of Claim 18 is also improper because Durham does not disclose step (c), which recites “if and only if the data structure encoded within the cookie data is determined to be valid in step (b), decoding the encoded data structure to reproduce the data structure within a memory of the server.” In connection with this claim language, the Examiner

cites col. 9, line 65 to col. 10, line 11 of Durham. Although the cited portion of Durham mentions that the cookie is dissembled and that its contents are used to generate a customized web page, nothing in the reference suggests that this process involves “decoding [an] encoded data structure to reproduce the data structure within a memory of the server.” In this regard, Durham is silent on how the decoded preference information is stored in the memory of the server.

Because Durham does not disclose all of the limitations of Claim 18, the anticipation rejections of Claims 18-21, 23 and 25-27 are improper and should be reversed.

Dependent Claim 22

The anticipation rejection of Claim 22 is improper because Claim 22 depends from Claim 18. In addition, the rejection of Claim 22 is improper because Durham does not disclose the limitations added by Claim 22, namely “wherein a plurality of data structures are encoded within the cookie data, and the method comprises determining which of the plurality of data structures should be fully decoded.” The portions of Durham relied on by the Examiner, namely col. 11, lines 15-36 and col. 15, lines 24-50, do not disclose a plurality of data structures encoded in cookie data, and do not disclose “determining which of the plurality of data structures should be fully decoded.”

Dependent Claim 24

The anticipation rejection of Claim 24 is improper because Claim 24 depends from Claim 18. In addition, the rejection of Claim 24 is improper because Durham does not disclose the limitations added by Claim 24, namely “wherein step (c) is performed by executable code that is not specific to a particular type of data structure.” The cited portion of Durham, namely col. 8, lines 5-20, simply does not disclose these limitations. As discussed above in connection with Claim 2, if the data elements encoded into cookies in Durham could somehow be considered “data structures,” Durham’s executable code would be specific to one or more particular types of data structures. This is because the types of data elements encoded into cookies in Durham are hard coded within the executable code.

Claims 28, 30, 32-34 and 36-38

Appellant is grouping together independent Claim 28, and Claims 30, 32-34 and 36-38 which depend from Claim 28, for purposes of argument. Appellant's arguments with respect to this group are based solely on independent Claim 28, which reads as follows:

28. A method of generating cookie data for storage on a computer of a user, the method comprising:

- identifying a set of data structures to be encoded within the cookie data;
- encoding the set of data structures within the cookie data according to schema data stored within a computer memory, said schema data specifying how the set of data structures is to be encoded within the cookie data; and
- incorporating into the cookie data at least one of the following to facilitate extraction of the set of data structures from the cookie data: (a) a schema identifier, (b) the schema data.

The anticipation rejection of Claim 28 (and thus Claims 30, 32-34 and 36-38) is improper because, among other reasons, Durham does not disclose "identifying a set of data structures to be encoded within the cookie data." In connection with this claim language, the Examiner points to Fig. 2, and col. 7, lines 21-50 of Durham. These portions of Durham describe how the preferences specified by a user are both written to a database and incorporated into a cookie. Nothing in Durham, however, suggests that this process involves "identifying a set of data structures to be encoded within the cookie data" as claimed. In this regard, the mere fact that Durham incorporates preference data read from a database into a cookie does not mean that Durham encodes data structures from this database into the cookie. To the contrary, as discussed above, Durham encodes this preference information at the preference bit level. In connection with this issue, Appellant submits that "data structure" is properly construed in Claim 28 to mean a set of data elements arranged according to an organizational scheme that is not the result of the cookie generation process.

The anticipation rejection of Claim 28 is also improper because Durham does not disclose "encoding the set of data structures within the cookie data according to schema data stored within a computer memory, said schema data specifying how the set of data structures is to be encoded within the cookie data." As mentioned above, Durham's encoding method is hard-coded within executable code, and not specified by schema data as claimed. (As mentioned above, the Final

Office Action does not clearly specify what the Examiner is treating as the recited “schema data.”) In addition, Durham does not encode a set of data structures within the cookie data as claimed. The portions of Durham cited by the Examiner (namely Figs. 4 and 5; col. 9, lines 39-40; col. 10, line 12 to col. 11, line 67; and Claim 5) do not suggest otherwise.

Because Durham does not disclose all of the limitations of Claim 28, the anticipation rejections of Claims 28, 30, 32-34 and 36-38 are improper.

Dependent Claim 31

The anticipation rejection of Claim 31 is improper because Claim 31 depends from Claim 28. In addition, the rejection of Claim 31 is improper because Durham does not disclose the limitations added by Claim 31, namely “wherein the set of data structures is encoded within the cookie data using executable code that is not specific to a particular type of data structure.” In connection with these limitations, the Examiner points to col. 8, lines 5-20. This portion of Durham describes steps that are performed when a client does not transmit a valid cookie. Nothing in this or any other portion of Durham discloses the encoding of a set of data structures within cookie data using executable code that is not specific to a particular type of data structure.

Dependent Claim 35

The anticipation rejection of Claim 35 is improper because Claim 35 depends from Claim 28. In addition, the rejection of Claim 35 is improper because Durham does not disclose the limitations added by Claim 35, namely “wherein the method is performed in an off-line mode to reduce a delay experienced by a user.” The text cited by the Examiner, namely, col. 9, line 65 to col. 10, line 11, does not disclose this feature. To the contrary, the cited text and the accompanying drawing (Fig. 2) indicate that the cookie is generated in response to a page request from a user, and is transmitted to the user’s computer with the corresponding response. See Fig. 2. In other words, Durham generates the cookie in an on-line mode, rather than an off-line mode as claimed.

Independent Claim 39

Claim 39 reads as follows:

39. A method for using browser cookies to store structured data, the method comprising:

storing schema data on at least one server computer of a web site system, said schema data specifying schemas used by executable software to (a) encode data structures within cookies for storage on user computers, and (b) decode said cookies to extract the data structures when the cookies are returned by the user computers; and

modifying the schema data over time to add data structures to, and remove data structures from, a set of data structures encoded within cookies by the executable software.

The anticipation rejection of Claim 39 is improper because, among other reasons, Durham does not disclose “storing schema data on at least one server computer of a web site system, such schema data specifying schemas used by executable software to (a) encode data structures within cookies for storage on user computers, and (b) decode said cookies to extract the data structures when the cookies are returned by the user computers.” The portions of Durham cited by the Examiner, namely Fig. 5, and col. 9, line 33 to col. 10, line 59, do not disclose these limitations. As explained above, Durham does not disclose the encoding of data structures within cookies. In addition, although Durham may use a schema to encode preference information into a cookie, the schema is embodied within the executable code itself, and not “schema data specifying schemas used by executable software” as claimed. The Final Office Action does not clearly specify what the Examiner is treating as the “schema data” recited in Claim 39.

The rejection of Claim 39 is also improper because Durham does not disclose “modifying the schema data over time to add data structures to, and remove data structures from, a set of data structures encoded within cookies by the executable software.” In connection with this claim language, the Examiner points to col. 7, lines 21-50 of Durham. This portion of Durham, however, says nothing about modifying schema data. As explained above, Durham’s use of a version number does not imply, or inherently involve, the use of schema data as claimed.

In response to Appellant’s arguments regarding the “modifying the schema data” language, the Examiner points out that Durham discloses modifying a cookie before sending it

back to the client. See Final Office Action at page 17, first full paragraph, citing col. 8, lines 1-5 of Durham. Nothing in the reference, however, suggests that the process of modifying the cookie involves modifying schema data, or that it involves adding or removing a data structure. Rather, Durham appears to merely be saying that additional preferences obtained from a user may be added to the cookie.

Because Durham does not disclose all of the limitations of Claim 39, the anticipation rejection of Claim 39 is improper and should be reversed.

Dependent Claim 40

The anticipation rejection of Claim 40 is improper because Claim 40 depends from Claim 39. In addition, the rejection of Claim 40 is improper because Durham does not disclose the limitations added by Claim 40, namely “wherein the schema data is modified according to a set of rules to enable the executable software to decode cookies encoded using both past and present schemas.” In connection with these limitations, the Examiner points to col. 3, lines 5-20 of Durham, which states, in part, that “a cookie-version number can be embedded to indicate how a server should interpret an incoming cookie.” While this statement may indicate that different cookies may be encoded differently, it does not indicate that “schema data is modified according to a set of rules” as claimed.

Dependent Claim 41

The anticipation rejection of Claim 41 is improper because Claim 41 depends from Claim 40. In addition, the rejection of Claim 41 is improper because Durham does not disclose the limitations added by Claim 41, namely “wherein the rules are enforced by a management layer.” The “personalization process” described at col. 7, lines 22-29 (cited by the Examiner) does not meet the limitations of Claim 41 because, e.g., it does not enforce rules used to modify schema data.

Dependent Claim 42

The anticipation rejection of Claim 42 is improper because Claim 42 depends from Claim 39. In addition, the rejection of Claim 42 is improper because Durham does not disclose the limitations added by Claim 42, namely “wherein the executable software is not specific to a particular type of data structure.” The portion of Durham relied on by the Examiner, namely col.

8, lines 5-20, does not disclose executable software that encodes data structures in cookies yet is not specific to a particular type of data structure.

2. Obviousness rejection of Claim 16 over Durham in view of McDonough

Claim 16 depends from Claim 1, and stands rejected on obviousness grounds over Durham in view of McDonough. The obviousness rejection of Claim 16 is improper because, among other reasons, Durham and McDonough et al. do not teach or suggest all of the limitations of Claim 1. For example, Durham and McDonough do not teach or suggest “schema data stored in the memory of the server, said schema data specifying past and present schemas used to encode data structures into cookies stored on user computers.” For the reasons explained above in connection with Claim 1, Appellant respectfully disagrees with the Examiner’s assertion that Durham discloses these limitations.

3. Obviousness rejection of Claim 29 over Durham in view of Goldberg

Claim 29 depends from independent Claim 28, and stands rejected on obviousness grounds over Durham in view of Goldberg. The obviousness rejection of Claim 29 is improper because, among other reasons, Durham and Goldberg do not teach or suggest all of the limitations of Claim 28. For example, Durham and Goldberg do not teach or suggest “identifying a set of data structures to be encoded within the cookie data,” or “encoding the set of data structures within the cookie data according to schema data.” For the reasons explained above in connection with Claim 28, Appellant respectfully disagrees with the Examiner’s assertion that Durham discloses these limitations.

Appl. No. : 10/694,509
Filed : October 27, 2003

VIII. CONCLUSION

For the reasons set forth above, Appellant respectfully submits that the rejections of Claims 1-42 are improper, and requests that these rejections be reversed.

Respectfully submitted,

KNOBBE, MARTENS, OLSON & BEAR, LLP

Dated: February 14, 2007

By: /RJS38297/
Ronald J. Schoenbaum
Reg. No. 38,297
Attorney of Record
2040 Main Street, 14th Floor
Irvine, CA 92614
(949) 721-2950

CLAIMS APPENDIX

1. A system for servicing web page requests, the system comprising:
a server that responds to user requests for web pages, said server comprising a memory;
schema data stored in the memory of the server, said schema data specifying past and present schemas used to encode data structures into cookies stored on user computers;
a conversion component executed by the server, said conversion component configured to use the schema data to identify and decode the data structures encoded within cookies received from user computers to generate temporary data structures within the memory of the server; and
application code executed by the server, said application code configured to use the temporary data structures to customize web pages requested by the user computers.
2. The system of Claim 1, wherein the conversion component is not specific to a particular type of data structure.
3. The system of Claim 1, wherein the conversion component supports a plurality of different types of data structures.
4. The system of Claim 1, wherein the schema data specifies the content and format of each of a plurality of data structures that are encoded within the cookies.
5. The system of Claim 1, wherein the schema data includes, for a given data structure that is encoded within cookies, an identification of primitive fields of the given data structure and datatypes of said primitive fields.
6. The system of Claim 5, wherein the schema data further includes address offsets of the primitive fields.
7. The system of Claim 1, wherein the schema data includes, for a given data structure that is encoded within cookies, an indication of a range of schema versions for which the data structure is valid, wherein the conversion component uses said range to determine whether a particular data structure encoded within a cookie is valid.

8. The system of Claim 1, wherein the schema data includes information about at least one data structure that is no longer in use.

9. The system of Claim 1, wherein the conversion component uses the schema data to determine which of a set of the data structures encoded within a received cookie are to be decoded for use.

10. The system of Claim 1, wherein the conversion component uses the schema data to determine whether a particular data structure that is encoded within a received cookie is to be decoded for use.

11. The system of Claim 1, wherein the conversion component is an executable function that is called by the application code.

12. The system of Claim 1, wherein the schema data is cached with random access memory of the server.

13. The system of Claim 1, wherein the schema data is stored within a file in the memory of the server.

14. The system of Claim 1, wherein the schema data is stored within a table in the memory of the server.

15. The system of Claim 1, wherein the conversion component uses a checksum included within a received cookie to evaluate whether the cookie has been modified.

16. The system of Claim 1, wherein the conversion component applies a decryption algorithm to encrypted information contained in the cookies received from the user computers.

17. The system of Claim 1, further comprising an encoding component that encodes data structures into cookies according to a current schema specified by the schema data.

18. A method of extracting information from cookies, the method comprising:

(a) receiving, at a server, cookie data that has a data structure encoded therein, said cookie data received from a user computer;

(b) determining whether the data structure encoded within the cookie data is valid, wherein the validity of the data structure is determined using information that identifies types of encoded data structures that are currently valid; and

(c) if and only if the data structure encoded within the cookie data is determined to be valid in step (b), decoding the encoded data structure to reproduce the data structure within a memory of the server.

19. The method of Claim 18, wherein the validity of the data structure as determined in step (b) reflects whether the data structure is currently used by any applications running on the server.

20. The method of Claim 18, wherein step (b) comprises identifying a version of a schema used to encode the data structure within the cookie data.

21. The method of Claim 18, wherein step (b) comprises using schema data stored within the memory of the server to determine whether the data structure is valid.

22. The method of Claim 18, wherein a plurality of data structures are encoded within the cookie data, and the method comprises determining which of the plurality of data structures should be fully decoded.

23. The method of Claim 18, further comprising using the data structure as reproduced in step (c) to customize a web page requested by the user computer.

24. The method of Claim 18, wherein step (c) is performed by executable code that is not specific to a particular type of data structure.

25. The method of Claim 18, wherein the data structure includes primitives of at least one of the following data types: 16-bit integers, 32-bit integers.

26. A server system that performs the method of Claim 18.

27. A computer program that embodies the method of Claim 18 stored within a computer readable medium.

28. A method of generating cookie data for storage on a computer of a user, the method comprising:

identifying a set of data structures to be encoded within the cookie data;

encoding the set of data structures within the cookie data according to schema data stored within a computer memory, said schema data specifying how the set of data structures is to be encoded within the cookie data; and

incorporating into the cookie data at least one of the following to facilitate extraction of the set of data structures from the cookie data: (a) a schema identifier, (b) the schema data.

29. The method of Claim 28, wherein the method comprises incorporating the schema data into the cookie data.

30. The method of Claim 28, wherein the method comprises incorporating a schema identifier into the cookie data, said schema identifier identifying one of a plurality of schemas used over a period of time to encode data structures within cookie data.

31. The method of Claim 28, wherein the set of data structures is encoded within the cookie data using executable code that is not specific to a particular type of data structure.

32. The method of Claim 28, wherein the set of data structures includes non-character primitives.

33. The method of Claim 28, wherein the set of data structures includes primitives of at least one of the following data types: 16-bit integers, 32-bit integers.

34. The method of Claim 28, further comprising incorporating a checksum into the cookie data to permit subsequent detection of whether the cookie data has been modified.

35. The method of Claim 28, wherein the method is performed in an off-line mode to reduce a delay experienced by a user.

36. A computer-readable medium comprising cookie data generated according to the method of Claim 28.

37. A computer system configured to perform the method of Claim 28.

38. A computer-readable medium that stores a computer program embodying the method of Claim 28.

39. A method for using browser cookies to store structured data, the method comprising:

storing schema data on at least one server computer of a web site system, said schema data specifying schemas used by executable software to (a) encode data structures within cookies for storage on user computers, and (b) decode said cookies to extract the data structures when the cookies are returned by the user computers; and

modifying the schema data over time to add data structures to, and remove data structures from, a set of data structures encoded within cookies by the executable software.

40. The method of Claim 39, wherein the schema data is modified according to a set of rules to enable the executable software to decode cookies encoded using both past and present schemas.

41. The method of Claim 40, wherein the rules are enforced by a management layer.

42. The method of Claim 39, wherein the executable software is not specific to a particular type of data structure.

Appl. No. : 10/694,509
Filed : October 27, 2003

EVIDENCE APPENDIX

None

RELATED PROCEEDINGS APPENDIX

None

3165371_1
120506